Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/114/2024.18276

Large Language Models: Development in Model Scale and Challenges

Zhouquan Lu

School of Foreign Languages, Shanghai Jiao Tong University, Shanghai, 200082, China

ruaa24@SJTU.edu.cn

Abstract. Since the 1950s, language modeling (Language Models, LMs) has been one of the primary approaches for tasks such as machine translation, as well as language understanding and processing. It has been widely applied in the field of natural language processing (Natural Language Processing, NLP), significantly improving the performance of tasks related to natural language understanding and generation. In recent years, large language models (LLMs) have made remarkable advancements in technical architecture and model scale, providing strong technological support for NLP and other fields. This paper presents a comprehensive review of the technological architecture and the development of the scale of large language models (LLMs), and delves deeply into the challenges these developments pose, along with the current strategies to address them. Finally, the paper summarizes and offers a prospective outlook on the future development directions of LLMs in terms of scale, providing insights and inspiration for the future development, training, and application of LLMs.

Keywords: Large Language Models (LLMs), Model Scale, Model Performance.

1. Introduction

As early as the 1950s, scientists began attempting to simulate the process of human language understanding to study linguistic rules, leading to the emergence of language models. With the advancement of computational power, expansion of datasets, and development of deep learning technologies, the strong predictive capabilities of neural networks aligned well with the discrete, high-dimensional nature of NLP, enabling language models to evolve from purely statistical models to neural network models. This marked a significant breakthrough in the field of NLP.

Unlike Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), the Transformer model[1], which introduced the attention mechanism, demonstrated exceptional performance in NLP tasks such as machine translation. In recent years, numerous pre-trained language models (PLMs) based on the Transformer architecture have emerged, such as the GPT series. The architectures of PLMs have become increasingly complex, and the number of model parameters has surged from 1 billion to the scale of hundreds of billions. Research indicates that there is a positive correlation between the scale of PLMs and their performance[2]. As the number of parameters grows exponentially, the models exhibit significant improvements in performance, understanding ability, and generalization capabilities.

However, the continuous expansion of model scale has introduced challenges such as insufficient computational resources, decreased inference efficiency, overfitting, and memory limitations. At the current stage, the increasing computational demands and training costs have become significant barriers to further enhancing model performance. To address these issues, numerous studies have focused on optimizing or compressing model architectures without compromising performance. Techniques such as knowledge distillation and quantization have been explored, along with methods to improve inference speed.

In the future, large language models will not be limited to text understanding and generation but will integrate with multimodal learning and multitask learning, extending their application to embodied agents. The computational demands of these models will continue to grow. To address the future opportunities and challenges of large language models, it is essential to thoroughly consider the development of model scale and its associated challenges. In this context, this paper first provides a brief overview of the development of LLMs' technical architecture and summarizes the evolution of model scale both domestically and internationally, discussing the relationship between model scale and performance. Next, it examines the challenges posed by expanding model scale. Subsequently, the paper elaborates on current technical research addressing these challenges. Finally, it offers an outlook on the future development of LLMs in terms of scale.

2. LLMs Technical Architecture

In 2017, Google innovatively introduced a feature extractor, Transformer[1], which incorporated the attention mechanism, breaking through the bottleneck of using RNNs and other neural networks for NLP tasks. The attention mechanism introduced in the Transformer architecture employs three feature vectors: Q (Query), K (Key), and V (Value). It calculates the attention weights for V by aggregating the attention between Q and K. Since its introduction, Transformer has become the foundational building block for nearly all LLMs today.

Large language models (LLMs) refer to language models pre-trained on massive amounts of textual data, capable of generating human-like text, answering questions, and completing other NLP tasks with high accuracy. To achieve this level of performance, LLMs are trained on text corpora containing hundreds of billions of tokens sourced from books, web pages, and other materials, forming pre-trained models. These models are further refined through instruction tuning, reward functions, and reinforcement learning-based alignment techniques to optimize their performance.

Currently, the mainstream architectures of LLMs can be categorized into three types: Encoder-only, Decoder-only, and Encoder-decoder. All three frameworks use unsupervised learning methods to perform predictive learning on datasets to improve performance. However, each framework focuses on different NLP tasks based on the specific unsupervised learning approach employed, resulting in the development of various LLMs based on these distinct architectures.

3. LLMs Model Scale

3.1. Development of Model Scale

Research indicates that the expansion of LLMs' model scale generally enhances model performance. Increased parameter counts enable the model to capture more textual features, thereby improving its comprehension and generalization abilities, which in turn leads to better performance across various tasks. Reviewing the development trajectory of LLMs, all models have scaled up during iterations, with parameter counts growing exponentially. Table 1 summarizes the parameter count evolution of large language models (LLMs) both domestically and internationally in recent years.

3.1.1. Development of International Models

In 2018, OpenAI introduced the GPT-1[3], a large language model based on the Transformer architecture and a Decoder-only framework. GPT-1 utilized unsupervised generative pretraining combined with supervised fine-tuning to tackle single-sequence text generation tasks, such as language inference and question answering. During unsupervised pretraining, GPT-1 stacked 12 Transformer layers, expanded the feedforward hidden layer to 3072 dimensions, and achieved a total parameter count of 117 million. Starting with GPT-2[4], the model abandoned supervised fine-tuning, expanded training data and network layers, and leveraged contextual learning capabilities to achieve full coverage of supervised learning tasks, overcoming GPT-1's limitations in language understanding and generation. GPT-2 increased the number of Transformer layers to 48 and the hidden layer dimensions to 1600, with parameters growing from 117 million to 1.5 billion. In the subsequent version, GPT-3[5], the parameter count surged to 175 billion, and the number of Transformer layers further increased to 96. It is estimated that the latest large language model, GPT-4[6], released in 2023, has parameter counts in the trillion range. With its massive scale, GPT-4 demonstrates powerful performance in text and multimodal tasks.

In the same year as GPT-1, Google released BERT[7], a bidirectional Transformer language model based on the Encoder-only framework. BERT had 24 Transformer layers and 340 million parameters. By 2022, Google had successively introduced LLMs with increasing parameter counts, including LaMDA[8], a dialogue-focused model with 137 billion parameters, and the ultra-large-scale PaLM[9], which reached 540 billion parameters.

The LLaMA[10] series, developed by Meta AI and released in 2023, improved performance over contemporary LLMs by expanding training data. The LLaMA series includes versions with parameter counts ranging from 7 billion to 65 billion, with the updated LLaMA2 increasing the maximum parameter count to 70 billion.

3.1.2. Development of Domestic Models

With the rapid growth of artificial intelligence, domestic institutions in China have kept pace by developing large language models tailored to the Chinese linguistic context, striving to achieve performance on par with international standards. The widespread adoption of ChatGPT in 2023 spurred explosive growth in the development of domestic LLMs. As of July 2023, China had released 130 large language models[11]. Among these, Baidu's early version of "Wenxin Yiyan" (Ernie Bot) featured 250 billion parameters, with 85% of its training corpus in Chinese. Alibaba's "Tongyi Qianwen" series included models with parameter counts ranging from 7 billion to 70 billion, equivalent to GPT-3 in overall scale. This demonstrates that the scale of domestic LLMs is now comparable to that of international counterparts.

Table1. Comparison of Parameter Counts in Recent Domestic and International LLMs

models	Release time	developers	Parameters/10^8
GPT	2018	OpenAI	1.17
BERT	2018	Google	3.40
GPT-2	2019	OpenAI	15.00
GPT-3	2020	OpenAI	1750.00
GLaM	2021	Google	1200.00
LaMDA	2022	Google	1370.00
PaLM	2022	Google	5400.00
LLaMA	2023	MetaAI	70.00-650.00
LLaMA2	2023	MetaAI	70.00-700.00
GPT-4	2023	OpenAI	—
ERNIE Bot	2023	Baidu	2500.00
Tongyi.ai	2023	Ali	70.00-700.00
Baichuan	2023	Baichuan AI	70.00

3.2. Model Scale and Model Performance

The relationship between model scale and model performance is complex and multifaceted. Research has revealed the phenomenon of emergent abilities[12], which refers to unique capabilities that do not exist in smaller models but emerge as the model scale increases. These abilities cannot be predicted by analyzing the limitations of smaller models. For instance, the few-shot prompting capability, extensively used in GPT-3, can only function effectively in sufficiently large models, allowing them to answer questions correctly using prompts without additional pretraining. While the expansion of model scale is a key condition for the emergence of such abilities, the underlying mechanisms remain unexplained. Investigating how emergent abilities arise and whether future expansions in model scale will lead to new capabilities provides a fresh perspective on enhancing model performance, representing a significant research direction in NLP.

Additionally, model performance exhibits a nonlinear relationship with key model parameters such as model scale, training data size, and computation amount, described as power law scaling[13]. This relationship indicates that during model pretraining, the loss L tends to decay in a power law pattern relative to resources like the number of parameters (N), the number of training samples (P), and the computational budget (C). By observing and analyzing this power law relationship, researchers can effectively adjust parameters, training data, and other resources to guide model design and training, achieving optimal performance within available resources. For example, during model pretraining, increasing the parameter count by two orders of magnitude may only reduce the loss by 0.5 points[13]. In such cases, developers should explore alternative methods to improve model performance rather than continuously scaling up the model.

4. Challenges Brought by Model Development

In recent years, the continuous expansion of LLMs has led to models like GPT-4 reaching parameter counts in the trillions. While this demonstrates that increasing model scale significantly enhances model performance, it also introduces a range of challenges that need to be addressed when scaling models indefinitely.

4.1. Dramatic Increase in Training Costs

From pretraining LLMs to subsequent inference processes, significant computational resources are required. The increase in parameters and training data further amplifies the demand for computational power. Figure 1 illustrates the evolution of parameter counts and computational requirements for various major models[14].

Taking GPT-3 as a reference, a single training run for GPT-3 requires approximately 1.75*10^{23} floating-point operations. Research indicates that during usage, ChatGPT consumes about 4874.4 PFlop/s-day[11]. Under such immense computational demands, both the training time and associated costs have grown exponentially compared to earlier models.

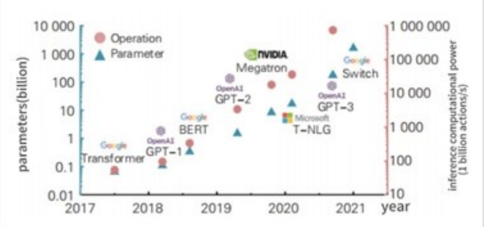

Figure 1. Model Parameter Counts and Computational Requirements

4.2. Decline in Inference Speed

As model scale continues to expand, training increasingly occupies large amounts of GPU memory. This reduces the model's ability to efficiently capture data features, resulting in slower inference speeds and hindering the model’s ability to deeply understand the characteristics of the training data.

4.3. Overfitting

The increase in parameter counts in LLMs enhances the model's ability to understand training datasets. However, excessive model scale can lead to over-reliance on training data, causing overfitting and a decline in generalization capability. Overfitting can result in risks such as reduced data authenticity and biases[15], which impact the fairness of model outputs. Preventing overfitting is a critical concern during the model training process and requires particular attention.

5. Technical Advances Addressing Challenges

Since the rapid development of large language models (LLMs), numerous technological approaches have emerged to maintain model performance while managing training costs and improving inference speed. These efforts have achieved significant progress. Below are the key existing techniques and innovations addressing training cost and inference speed challenges.

5.1. Mixture of Experts Training

The core idea of Mixture of Experts (MoE) training is to enhance model performance by decomposing tasks and utilizing multiple expert networks to process relevant data separately. MoE, essentially a Transformer model, increases the model’s "sparsity," allowing for faster training speeds while requiring fewer computational resources. For instance, Google’s multimodal large language model Gemini 1.5 employs the MoE framework to distribute task requests to smaller "expert" neural networks, thereby improving response speed. However, MoE faces the challenge of parameter inefficiency. This is due to the independence of different layers in the model, which fails to leverage historical information during training. Recent research by Qiu et al. on RmoE[16] introduces the concept of inter-layer routing decision dependencies. By using GRUs (Gated Recurrent Units) to connect consecutive layers, RmoE achieves efficient parallel computation and addresses the issue of parameter redundancy.

5.2. Quantization

Model quantization focuses on converting floating-point parameters into integers to reduce computational demands. This approach decreases memory usage, lowers power consumption, and enhances inference speed on hardware optimized for integer computation, effectively addressing challenges arising from model scaling. For LLMs, two primary quantization methods are currently prevalent: coarse-grained quantization and fine-grained quantization. The former quantizes the entire tensor or along a specific dimension, while the latter slices a dimension into multiple parts and quantizes each slice individually.

From an application perspective, post-training quantization is widely used to reduce model storage and computational complexity. Post-training quantization compresses computational demands by quantizing model weights, often employing fine-grained quantization methods without requiring calibration datasets or altering the model architecture. Additionally, post-training quantization can quantize activation values within the model. Recent research has extensively explored methods to minimize quantization errors and determine the required precision for models. For example, Zhang et al. proposed DGQ (Dual Grained Quantization)[17], which addresses the inefficiency of fine-grained quantization due to disrupted continuous matrix multiplications. DGQ adopts an A8W4 quantization strategy (using INT8 for activations and INT4 for weights) and performs matrix multiplication through an INT8 kernel, thereby improving inference efficiency.

5.3. Knowledge Distillation

Knowledge distillation is primarily used for large language models (LLMs) that require substantial computational resources and cannot operate on devices with limited capabilities, such as mobile devices. This technique follows a "teacher-student" model, where a larger, higher-performing "teacher" model transfers knowledge to a simpler "student" model. The goal is to enable the "student" model to replicate the "teacher" model's capabilities within limited computational resources while enhancing the student's generalization abilities.

In recent years, knowledge distillation has made significant advancements and has been widely applied in fields like computer vision and natural language processing. The approach has evolved from initial output distillation to feature distillation and relational distillation, which delve deeper into the internal relationships and training processes of the "teacher" model. However, the technique still exhibits shortcomings in some applications. Research by Agarwal et al.[18] highlights a distribution mismatch between the sequences output by the "teacher" and those generated by the "student" during training in autoregressive sequence models. To address this, the study introduced Generalized Knowledge Distillation (GKD), which allows the "student" model to learn from its self-generated output sequences while leveraging feedback from the "teacher" model to enhance performance.

5.4. Pruning

Pruning, as its name suggests, refers to the removal of neurons or connections below a certain threshold to increase the sparsity of a model. The primary aim is to reduce the size, complexity, and computational demands of the model, while simultaneously improving inference speed and reducing energy consumption. In the domain of compressing large language models (LLMs), pruning techniques are generally classified into two categories: structured pruning and unstructured pruning. Structured Pruning: This method involves pruning specific structures, such as neurons, channels, or layers, while maintaining the overall architecture of the model. This ensures compatibility with hardware acceleration, making it practical for implementation. Unstructured Pruning: In contrast, this method independently removes individual parameters without adhering to the model’s predefined structure. While it achieves a high compression ratio, it cannot leverage existing hardware architectures for acceleration, limiting its practicality. Although pruning effectively improves inference speed and reduces model size, it often results in some degree of performance degradation, adhering to the power law scaling principle. To address this limitation, Sorscher et al.[13] introduced data pruning, which breaks the constraints of the power law scaling principle. The study focuses on identifying optimal data pruning metrics to determine the order in which training samples are discarded. By carefully selecting the pruned dataset size, the method achieves exponential scalability while maintaining robust performance.

6. Conclusion

To enhance model performance, the parameter count of LLMs has grown rapidly, following the power law scaling principle. As of now, LLMs have reached parameter counts in the trillion range, nearing saturation in their applications. Continuing to scale up model parameters for optimization requires immense computational resources and significant training costs, while also introducing challenges such as slower inference speeds and potential ethical concerns. To address the challenges posed by excessive model scale, techniques such as pruning, knowledge distillation, quantization, and other model optimization approaches like Mixture of Experts (MoE) have been developed. These techniques aim to effectively mitigate issues such as high training costs. However, each of these methods still faces various limitations in practical application, necessitating further research in the future.

LLMs are continually evolving, unlocking more capabilities. In the future, balancing model scale and performance will remain a critical area of research, reflected in two key aspects:

(1) At present, LLMs primarily serve as foundational models focused on language-related tasks. However, integrating LLMs with multimodal, multitask, and cross-domain capabilities to broaden their application scope is an inevitable trend. In this context, model scale will need to expand to improve multimodal integration and comprehension capabilities across different modalities. Developing methods to compress model scale and reduce computational demands under these conditions will be a key focus for future research.

(2) Driven by the vision of using LLMs to interact with real-world environments, developing agents based on LLMs has become a prominent direction in artificial intelligence. These agents can operate in virtual environments, such as mobile devices or computer web pages, utilizing the powerful reasoning capabilities of LLMs to perform tasks like online shopping or sending emails. However, due to device

limitations, the LLMs employed in such agents must undergo structural optimization, continuously seeking a balance between scale and performance.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2023). Attention is all you need. arXiv preprint arXiv:1706.03762. https://doi.org/10.48550/arXiv.1706.03762

[2] Arora, K., & Rangarajan, A. (2016). Contrastive entropy: A new evaluation metric for unnormalized language models. arXiv preprint arXiv:1601.00248. https://doi.org/10.48550/arXiv.1601.00248

[3] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (n.d.). Improving language understanding by generative pre-training. Unpublished manuscript.

[4] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (n.d.). Language models are unsupervised multitask learners. Unpublished manuscript.

[5] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165. https://doi.org/10.48550/arXiv.2005.14165

[6] OpenAI, et al. (2024). GPT-4 technical report. arXiv preprint arXiv:2303.08774. https://doi.org/10.48550/arXiv.2303.08774

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171–4186). Minneapolis, MN: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

[8] Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., ... & Le, Q. V. (2024). LaMDA: Language models for dialog applications. arXiv preprint arXiv:2201.08239. Retrieved August 14, 2024, from https://arxiv.org/abs/2201.08239v3

[9] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Dean, J. (2022). PaLM: Scaling language modeling with Pathways. arXiv preprint arXiv:2204.02311. https://doi.org/10.48550/arXiv.2204.02311

[10] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288. Retrieved August 14, 2024, from http://arxiv.org/abs/2307.09288

[11] Wang, Y., Li, Q., Dai, Z., & Xu, Y. (2024). Research status and trends of large language models. Journal of Engineering Science, 46(8), 1411–1425. https://doi.org/10.13374/j.issn2095-9389.2023.10.09.003

[12] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... & Le, Q. (2022). Emergent abilities of large language models. arXiv preprint arXiv:2206.07682. https://doi.org/10.48550/arXiv.2206.07682

[13] Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., & Morcos, A. S. (2023). Beyond neural scaling laws: Beating power law scaling via data pruning. arXiv preprint arXiv:2206.14486. https://doi.org/10.48550/arXiv.2206.14486

[14] He, S., Mu, C., & Chen, C. (2024). Hardware architecture for large language models based on in-memory computing chips. ZTE Communications, 30(2), 37–42. https://doi.org/10.12142/ZTETJ.202402006

[15] Weidinger, L., Uesato, J., Michel, B., Chughtai, T., Dathathri, S., Bauer, M., ... & Glaese, A. (2021). Ethical and social risks of harm from language models. arXiv preprint arXiv:2112.04359. https://doi.org/10.48550/arXiv.2112.04359

[16] Qiu, Z., Du, S., Lin, J., Yang, X., & Song, Y. (2024). Layerwise recurrent router for mixture-of-experts. arXiv preprint arXiv:2408.06793. https://doi.org/10.48550/arXiv.2408.06793

[17] Zhang, L., Fei, W., Wu, W., He, Y., Lou, Z., & Zhou, H. (2023). Dual grained quantization: Efficient fine-grained quantization for LLMs. arXiv preprint arXiv:2310.04836. https://doi.org/10.48550/arXiv.2310.04836

[18] Agarwal, R., Xie, Y., Soares, L. B., Raffel, C., Narang, S., Devlin, J., & Madaan, A. (2024). On-policy distillation of language models: Learning from self-generated mistakes. arXiv preprint arXiv:2306.13649. https://doi.org/10.48550/arXiv.2306.13649